21. Multiple-Sensor People Localization in an Office Environment

Gang Wei, Valery A. Petrushin, and Anatole V. Gershman

Summary. This chapter describes an approach for people localization and tracking in an office environment using a sensor network that consists of video cameras, infrared tag readers, a fingerprint reader, and a PTZ camera. The approach is based on a Bayesian framework that uses noisy, but redundant data from multiple sensor streams and incorporates it with the contextual and domain knowledge that is provided by both the physical constraints imposed by the local environment where the sensors are located and by the people who are involved in the surveillance tasks. The experimental results are presented and discussed.

21.1 Introduction

The proliferation of a wide variety of sensors (video cameras, microphones, infrared badges, RFID tags, etc.) in public places such as airports, train stations, streets, parking lots, hospitals, governmental buildings, shopping malls, and homes has created the infrastructure that allows the development of security and business applications. Surveillance for threat detection, monitoring sensitive areas to detect unusual events, tracking customers in retail stores, controlling and monitoring movements of assets, and monitoring elderly and sick people at home are just some of the applications that require the ability to automatically detect, recognize, and track people and other objects by analyzing multiple streams of often unreliable and poorly synchronized sensory data. A scalable and robust system built for this class of tasks should also be able to integrate this sensory data with contextual information and domain knowledge provided by both the humans and the physical environment to maintain a coherent and logical picture of the world over time. While video surveillance has been in use for decades, systems that can automatically detect and track people (or objects) in multiple locations using multiple streams of heterogeneous and noisy sensory data is still a great challenge and an active research area. Since the performance of these automatic systems is not at the level at which they can work autonomously, there are human experts who are still part of the loop. It is important to develop techniques that can help human experts in this task by organizing and presenting the video surveillance data in a summarized manner, and highlighting unusual or rare events for further research by the experts. Many approaches have been proposed for object tracking in recent years.

They differ in various aspects such as number of cameras used, type of cameras and their speed and resolution, type of environment (indoors or outdoors), area covered (a room or a hall, a hallway, several connected rooms, a parking lot, a highway, etc.), and location of cameras (with or without overlapping fields of view). Some of the approaches are reviewed below. However, the performance of most systems is still far from what is required for real-world applications.

The objective of our research is to bridge the gap between the needs of practical applications and the performance of current surveillance algorithms. We seek solutions in the following directions:

- Developing a framework for logical integration of noisy sensory data from multiple heterogeneous sensory sources that combines probabilistic and knowledge-based approaches. The probabilistic part is used for object identification and tracking, and the knowledge-based part is used for maintaining overall coherence of reasoning.
- Exploiting local semantics from the environment of each sensor. For example, if a camera is pointed at a location where people usually tend to stand, the local semantics enable the system to use the "standing people" statistical models, as opposed to a camera pointing at an office space where people are usually sitting.
- Taking advantage of data and sensor redundancy to improve accuracy and robustness while avoiding the combinatorial explosion.
- Taking advantage of human guidance when it is available.
- Developing approaches and tools for efficient event clustering, classification, and visualization.
- Developing robust and scalable systems that work in real environments.

21.2 Environment

This research is a part of Multiple Sensor Indoor Surveillance (MSIS) project, which pursues the above-mentioned objectives. The backbone of the MSIS environment consists of 32 AXIS-2100 webcams, a pan-tilt-zoom (PTZ) camera, a fingerprint reader, and an infrared badge ID system (91 readers that are installed on the ceiling) that are sensing an office floor for Accenture Technology Labs in Chicago (Figure 21.1). The webcams and infrared badge system cover two entrances, seven laboratories and demonstration rooms, two meeting rooms, four major hallways, four open-space cube areas, two discussion areas, and an elevator area. Some areas are covered by multiple cameras, the maximum overlap being with up to four cameras. The total area covered is about 18,000 ft^2 (1,670 m^2). The fingerprint reader is installed at the entrance and used for matching an employee with his or her visual representation. The PTZ camera is watching the main entrance and northwestern cube area, and is used for face recognition.

Figure 21.2 presents the architecture of the system. It consists of three layers. The bottom layer deals with real-time image acquisition and feature extraction. It consists of several networked computers, with each computer running an agent that receives signals from 3 to 4 webcams, detecting "events," storing images for that

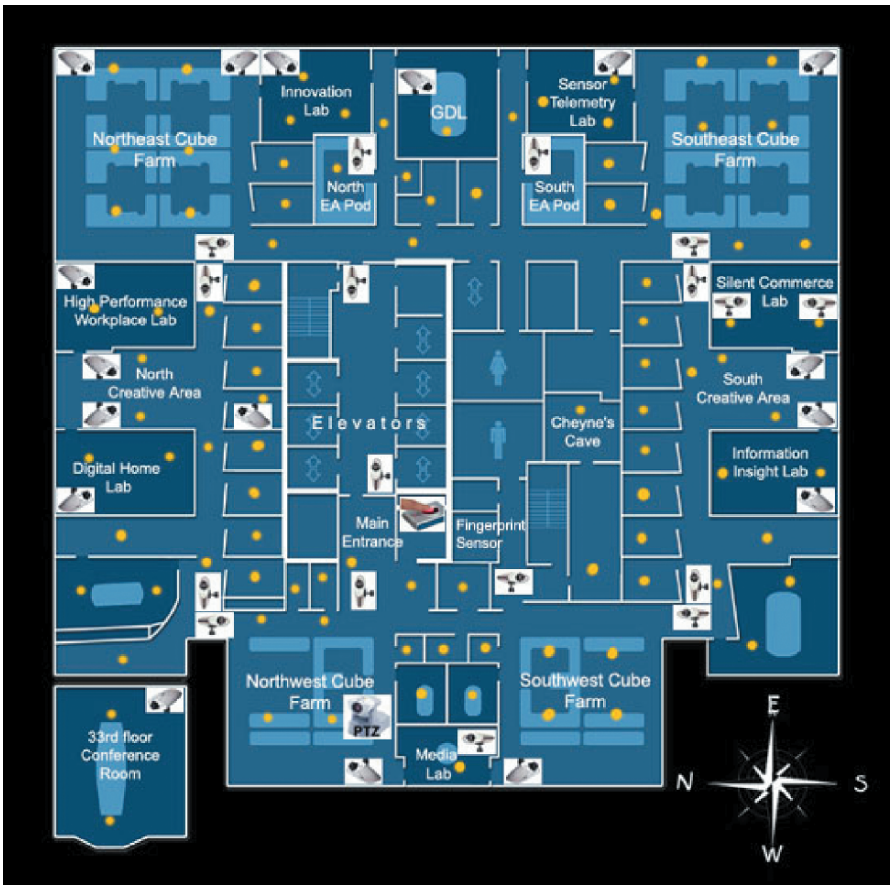

Fig. 21.1. Locations of Web cameras and infrared badge readers. Here IR badge readers are represented as dots, cameras are represented by small images that show their orientation, the PTZ camera has the label "PTZ" on the image, and the fingerprint reader is represented by a corresponding image near the main entrance.

event in the image repository in JPEG format, extracting features and saving them in the database. The event is defined as any movement in the camera's field of view. The average signal sampling frequency is about 3 frames per second. Three more agents acquire and save in the corresponding databases information about events detected by the infrared badge ID system, and the results of fingerprint and face recognition. The event databases serve as a common resource for applications of higher levels.

The middle layer consists of a set of application agents that use the features extracted at the bottom layer. The results of these agents go to the databases. Depending on the objective of the application it may use one, several, or all cameras and some other sensors.

The top layer consists of a set of meta-applications that use the results of the middle layer applications, integrate them, derive behavioral patters of the objects, and

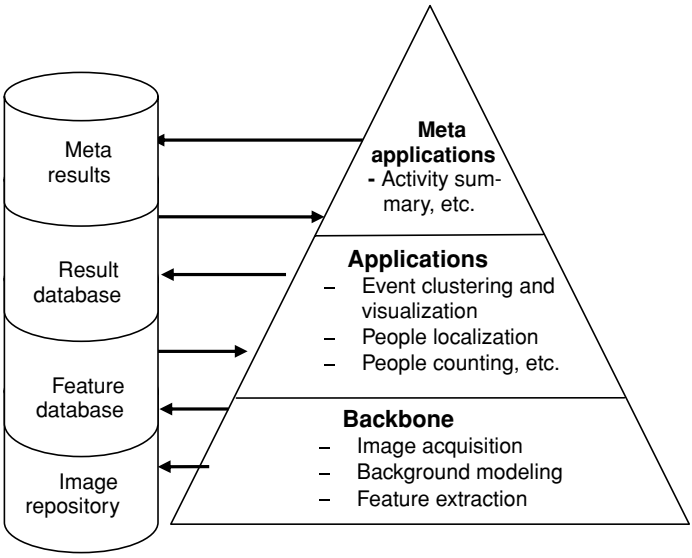

Fig. 21.2. General architecture of the Multiple Sensor Indoor surveillance system.

maintain the consistency of results. The applications of this layer are also responsible for maintaining the databases of the system and creating reports on its performance.

The MSIS project has the following objectives.

- Create a realistic multisensor indoor surveillance environment.
- Create an around-the-clock working surveillance system that accumulates data in a database for three consequent days and has a GUI for search and browsing.
- Use this surveillance system as a base for developing more advanced event analysis algorithms, such as people recognition and tracking, using collaborating agents and domain knowledge.

The following agents or applications have been considered.

- Creating a real-time image acquisition and feature extraction agent.
- Creating an event classification and clustering system.
- Search and browsing of the Event Repository database using a Web browser.
- Counting how many people are on the floor.
- Creating a people localization system that is based on evidence from multiple sensors and domain knowledge.
- Creating a real-time people tracking system that gives an optimal view of a person based on multiple cameras and prediction of person's behavior.
- Creating a system that recognizes a person at a particular location and interacts with him or her, for example, the system can send a voice message to the person and get his or her response.
- Creating a system that recognizes the behavior patterns of people.

- Creating a system that maintains the consistency of dynamic information about the events that was collected or derived by the other agents.

The above-mentioned applications are currently at different stages of completeness (see Chapter 5 in this book for a description of an application for event classification and clustering). This chapter describes the people localization system that is based on evidence from multiple sensors and domain knowledge.

21.3 Related Works

There are many papers devoted to video surveillance using single and multiple cameras. They differ in many aspects such as indoor/outdoor surveillance, people or/and vehicle tracking, using overlapping or nonoverlapping cameras, using mono or stereo, color or grayscale cameras, etc. Below we shall focus on some research that deals with indoor people identification and tracking.

A system described in work [1], which is a single camera system that was created for tracking people in subway stations, used the luminance contrast in YUV color space to separate people blobs from the background. The coordinates and geometric features of the blobs are estimated and two-way matching matrices algorithm has been used to track (overlapping) blobs.

In [2] one color static camera has been used to track people in indoor environment. It used several interacting modules to increase tracking robustness. The modules are a motion tracker that detects moving regions in each frame, a region tracker that tracks selected regions over time, a head detector that detects heads in the tracked regions, and an active shape tracker that uses models of people shape to detect and track them over time. The interaction among modules allows them dynamically incorporating and removing static objects into/from the background model, making prediction about a person's position and moving direction, and recovering after occlusions.

In the Microsoft's EasyLiving project [3] two color stereo cameras have been used for real-time identification and tracking up to three people in a rather small room (5 m by 5 m). The system evaluates 3D models of blobs and clusters them to fit a people-shaped blob model. Then the centroids of the blobs are projected into the room ground plan. The quantized RGB color histogram and histogram intersection are used for person's identity maintenance. A histogram is estimated for each person viewed by each camera in each visited cell of 10×10 grid of the floor plan. The person tracker module keeps the history of the person's past locations and uses it to predict current location. If the predicted location contains several candidates, then color histograms are used to disambiguate them. If no candidates found the system keeps unsupported person tracks active until new data arrive. For supported track their histories are updated and new predictions are calculated. In spite of low image processing rate (about 3.5 Hz) the system works well with up to three people, who are not moving too fast and not wearing similarly colored outfits.

The M2Tracker system [4] uses from 4 to 16 synchronized cameras to track up to six people walking in a restricted area (3.5 m by 3.5 m). The system identifies people using the following models for segmenting images in each camera view: color

models at different heights, presence probabilities along the horizontal direction at different heights, and ground plane positions tracked using a Kalman filter. Then the results of one camera segmentation are matched for pairs of cameras to estimate 3D models for each person and estimate the object location on the ground plane using Gaussian kernels to create location likelihood map. The system merges results from several pairs of cameras until the ground plane positions are stable. Then the current positions of people are updated, and new predictions are calculated. Because of high computational complexity, the system cannot work in real time, but the authors hope that code optimization efforts and advances in computing will make it possible in the future.

The system presented in [5] uses several nonoverlapping cameras and knowledge about topology of paths between cameras. It probabilistically models the chain of observation intervals for each tracked person using Bayesian formalization of the problem. To estimate the optimal chain of observation, the authors transform the maximum a posteriori estimation problem into a linear program optimization.

The approach proposed in [6, 7] uses multiple synchronized grayscale overlapping cameras for tracking people and selecting a camera that gives the best view. The system consists of three modules: single view tracking, multiple view transition tracking, and automatic camera switching. The system uses the following features for each person: locations of selected feature points, intensity of the selected feature points, and geometric information related to a coarse 2D human body model. The multivariate Gaussian models and Mahalonobis distances are used for people modeling and tracking. The class-conditional distribution for spatial and spatial–temporal matching is used for the multiple view transition tracking for matching predicted location and body model size. The automatic camera switching is necessary if the person is moving out of the current camera's field of view, or the person moves too far away, or the person is occluded by another person. The system selects a camera that will contain the person over the largest time accordingly the current prediction of the person's movement. The experiments with three cameras in various indoor environments showed high robustness of people tracking (96–98%).

The KNIGHTM system [8, 9] is a surveillance system that uses several overlapping and/or nonoverlapping uncalibrated color cameras for people tracking. The system uses spatial and color Gaussian probability distributions for each person to identify and track people in one camera view. The person identification is based on voting of foreground pixels. If two or more people receive essential percentage of votes from the same region, then the systems assume that partial occlusion of people happens. In case of complete occlusion a linear velocity predictor is used for disambiguation. To track people across multiple cameras, the system during the training period learns the field of view lines of each camera as viewed in the other cameras. This information and knowledge of cameras' location are used for identification of moving people. The experiments with three cameras and three different camera setups gave promising results.

The authors of the paper [10] suggest system architecture and scenarios of multiple camera systems that take advantage of recent achievements in video camera technology, such as omnidirectional and PTZ cameras. Using the combination of

such cameras allows creating an intelligent surveillance system that can automatically select an optimal camera view to track and recognize people and their behavior.

21.4 Feature Extraction

In this section we describe camera specification and our approach to extracting visual features that are used by all applications.

21.4.1 Camera Specification

We shall consider using for the surveillance task multiple static cameras with low frame sampling rate (3–5 Hz) which is typical for Web cameras. The advantages of indoor environments comparing to the outdoor ones are the following: there are no sharp shadows, illumination changes rather slow, speed of the objects is low, because the objects of interest are people. Besides, we can use our knowledge for specifying important areas (e.g., working places in cubicles) and unimportant areas (such as reflecting surfaces) in a camera's view. The disadvantages are that many places in an indoor environment are unobservable; people can easily change the direction of movement and the people can be often occluded by furniture or the other people.

Each camera has a specification that includes the following data.

- *Operating zone* is an area that is used for feature extraction. For some cameras, only part of their view area is worth to use. Having smaller operating zone expedites processing.
- *Background modeling type* sets the type of background modeling for the camera. The following background models are currently supported: single frame and median filtering. More information on background modeling can be found below.
- *Indicators* are some small areas and associated with them recognizers that allow detecting some local events such as light in an office is on/off, a door is open/closed, etc. The indicators play a double role—they can be used to improve the background modeling, and they are additional pieces of evidence about the state of the environment.
- *Important areas* are areas that the surveillance system pays special attentions, such as doorways, working places in cubicles, armchairs in a hall, etc.
- *Unimportant areas* are areas that must be ignored because they are sources of noise. Such areas are reflective surfaces, TV screens, computer monitors, etc.
- *Camera calibration data* is location of markers that allow estimating distances to objects in cameras' views. This data is used for estimating objects' location, their geometrical features and speed.

Figure 21.3 gives an example of camera specification. Here there are two indicators that detect such events as lights are on/off (in a meeting room on the left) and the door to the meeting room is open/closed. Areas of indicators are represented as black

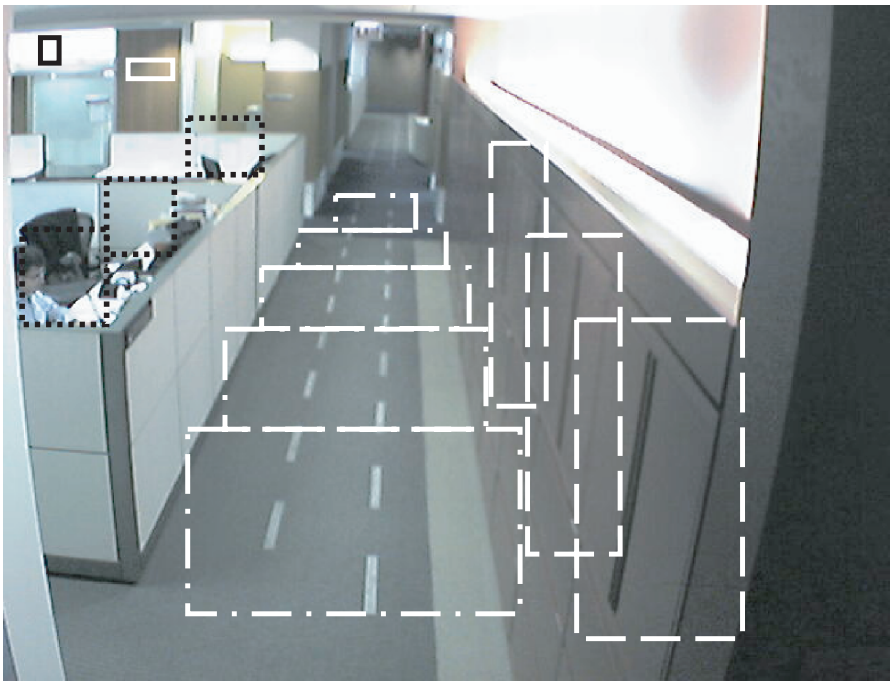

Fig. 21.3. Camera specification.

and white solid rectangles. The light indicator uses average intensity and a threshold as a recognizer, and the door indicator checks for a horizontal edge as a recognition that the door is open. Three black dotted rectangles on the left represent important areas—the working places in cubicles. Three white dashed rectangles on the right mark unimportant areas, which are surfaces that get shadow when a person is passing by. White markers on the floor are used for camera calibration. Five white dash-dotted rectangles split the walkway into zones. Ranges of eligible values for objects' height and width are associated with each zone and are used for blob extraction. A tool with a GUI has been designed to facilitate camera specification.

21.4.2 Background Modeling

The objective of background modeling techniques is to estimate value of pixels of the background frame, that is, the frame without any moving objects. Then this frame is subtracted pixel by pixel from a current frame to detect the pixels that belong to moving objects (foreground pixels). Many approaches for background modeling have been developed [11, 12].

The simplest approach is to use a single frame that is acquired when no motion or changes within a sequence of frames are detected. The single frame method works well when a camera watches a scene that has periods of time without moving objects,

for example, for a camera that is watching a hallway. To take into account the change of luminosity over time, the system has to periodically update the single frame model. The advantage of the single frame method is that it does not require any resources for maintaining the model. The disadvantage is that it does not work for scenes with intensive motion.

Another approach is to accumulate and maintain a pool of N frames, where N is an odd number. The value of each pixel of the background model is estimated as medians of the corresponding pixels of frames from the pool. This model is called the median filter background model. An alternative approach is to use mean instead of median; however, such model is sensitive to outliers and requires the large pool size N to be stable. The median filter model works well when each pixel of the scene is covered by moving objects less than 50% of time. The advantage of the median filter model is that it can be used for scenes with high motion, for example, for a camera that is watching cubicles. The disadvantage is that it requires additional memory for storing pool of frames and computations for maintaining the model [11].

The above models assume that each background pixel has only one value, but sometimes it is not true, for example, when a camera watches a bush or a branch of a tree that is shaken by wind; or a billboard that shows sequentially two advertisements. To model such backgrounds more advanced techniques are required. One of the approaches is to use Gaussian mixture models (GMM). In this approach each background pixel is modeled as a mixture of 3–5 Gaussians. Each Gaussian has a weight that is proportional to the frequency of pixels to have the value represented by the Gaussian. It is assumed that Gaussians that represent background values have higher weights and the sum of their weights reaches 0.6–0.8. Instead of subtracting background image from the current frame, each pixel of the current frame is checked for belonging to the background Gaussians if the probability is high it is marked as background, otherwise as foreground. Creating GMM for each background pixel requires intensive training, and changes in luminosity require dynamic adaptation of models. The advantage of the background modeling using GMM is that it works for periodically or stochastically changing environments. The disadvantages are that its success depends on training and parameter tuning, and it has high computational complexity and memory requirements [13, 14].

Real-time processing puts additional restrictions on background modeling. The real-time system cannot wait to accumulate training data, train the models, and then catch up by processing all postponed frames. It does not have enough processing power to implement such scenario. That is why we adopted an approach that uses two background modeling techniques and switches between them when it is needed. The system loops through the basic cycle that consists of the following steps: image acquisition, motion detection, if motion is not detected then the system is idle till the next cycle otherwise it processes the image, which includes extracting foreground pixels, extracting and storing features, and maintaining the background model. A sequence of cycles that have motion forms a *dynamic event*. The dynamic events are separated by events with no motion or *static events*.

If the camera uses the single frame model then the system starts by acquiring a single frame background during a static event and updates it not less than in T

Table 21.1. Using the single frame and adaptive models for real-time processing.

Model type	Initialization	Flicker	Local change of illumination	Global change of illumination
Single frame (SF)	Start a new SF	Skip frame	Patch the effected area	Start a new SF
Adaptive model (AM)	Start a new SF Accumulate data Generate and switch to AM		Generate a SF from AM Patch the effected area Accumulate data Generate and switch to AM	Start a new SF Accumulate data Generate and switch to AM

seconds by picking up a frame from a static event that lasts not less than D seconds. The parameters T and D can be specified for each camera (the default values are $T = 120$ and $D = 60$). There are three kinds of events that require attention for robust background modeling. The first one is a flicker that is a short abrupt change in illumination due to camera noise. The second kind of events is a local luminosity change; for example, lights get on or off in an office, which is a part of the scene. And the third kind of events is global luminosity change when more then 60% of pixels are changed; for example, when lights get on/off in the room, which is observed by the camera. The system reacts differently for each kind of events. When a flicker occurs, the system skips the frame. When a local luminosity change occurs, the system recognizes this case, using an indicator and patches the effected area with values extracted from the new frame. If the system uses an adaptive background model it first generates a single frame model and then patches the effected area. In case of global change the system acquires a new single frame. If the camera uses an adaptive background model, then the system acquires a single frame model and start accumulate data for creating a new adaptive model. It picks up frames from static events using parameters T and D. When the desired number of frames is reached the systems generates and switches to the adaptive model and begins to maintain it. Table 21.1 summarizes the system's behavior in different modes.

As an adaptive model we used the median filter model with a pool of size $N = 51$. Maintaining the model requires discarding the oldest frame, adding a new one and sorting the values for each pixel. The system using a version of a dynamic deletion–insertion algorithm to avoid sorting and improve the speed of model maintenance.

For dynamic events the background model is subtracted from current frame for detecting foreground pixels. Then some morphological operations are applied to remove noise and shadows. After this, the foreground pixels are separated into blobs using the calibration information for each camera. Finally a set of candidate blobs is selected for feature extraction.

21.4.3 Visual Feature Extraction

A person's most distinguishable visual identity is his or her face. However, in many practical applications the size and image quality of the face do not allow traditional face recognition algorithms to work reliably, and sometimes the human face is not visible at all. Therefore, our people localization system uses face recognition as an

auxiliary means that is applied for only some areas of some cameras. The other salient characteristic of a person are sizes of the body, color of the hair, and color of the clothes that is on the person. At any given day, a person usually wears the same clothes, and thus the color of person's clothes is consistent and good discriminator (unless everybody wears a uniform). We use color histograms in different color spaces as major features for distinguishing people on the basis of their clothes.

The blob is processing in the following manner. The top 15% of the blob, which represents the head, and bottom 20%, which represents the feet and also often includes shadow, are discarded and the rest of the region is used for feature extraction. We used the color histograms in RGB, normalized RGB (see Equations (21.1)–(21.2)), and HSV color spaces with the number of bins 8, 16, and 32.

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B} \tag{21.1}$$

$$l = \frac{R+G+B}{3} \tag{21.2}$$

where r, g, and b are red, green, blue components, and l is the luminance in the normalized RGB space, and R, G, and B are red, green, and blue components in the RGB space.

After some experiments we chose the 8-bin color histogram in the normalized RGB space for the r, g, and l components, which gave a good balance between computation efficiency and accuracy.

21.4.4 People Modeling

For modeling a person on the basis of his or her appearance, we use several approaches. The simplest one is to use all pixels of the blob for training a color histogram. Another approach is to fit a Gaussian or a Gaussian Mixture Model to the training data. A more elaborate person modeling includes two models—one for top and another for the bottom part of the body.

Let us assume that we built a model M_H for a human H. To estimate how well the data D extracted from a new blob R fits the model, we can consider the model as a probability density function and estimate the likelihood of the data set using Equation (21.3), which assumes that pixels' values are independent. The data set D can include all pixels of the blob or a randomly selected subset of particular length (usually 50–100 pixels are enough for reliable classification). In case when two (top and bottom) models are used for modeling, Equation (21.3) should be extended to include products of likelihoods for each model over corresponding data points. In practice, a log-likelihood function is used instead of likelihood one.

$$L(D \mid M_H) = \prod_{i=1}^{N} p(x_i \mid M_H) \tag{21.3}$$

where $x_i \in D$ are points of the data set D, $p(\cdot \mid M_H)$ is the probability distribution function for the model M_H, N is the number of points in the data set D.

The type of probability distribution function depends on the type of the model used. For example, in case of color histogram it can be approximated as a product of corresponding values for pixel's components. Below we shall use H instead of M_H if it does not cause any confusion.

21.5 People Localization

This section presents a Bayesian framework for people localization that allows the integration of evidence of multiple sensor sources. Our task is to localize and track N objects in a space of known geometry with stationary sensors of different kinds. The number of objects may change dynamically over time when an object arrives or leaves. The sensing zones for some sensors can overlap. We assume that there are two types of objects: known objects (employees) and unknown objects (guests or customers). The space is divided into "locations." Time is sampled into ticks. The tick duration is selected depending on the sampling frequencies of the sensors. It should be large enough to serve as a synchronization unit and small enough so that objects can either stay in the same location or move only to an adjacent one within a tick.

21.5.1 Sensor Streams

Each object is represented by a set of features extracted from sensor streams. We are currently using four sources of evidence.

21.5.1.1 Video Cameras

This is very rich data source, but requires a lot of sophisticated processing to extract useful information. Processing this source requires solving problems such as background modeling, object tracking, occlusion resolution, and object recognition. Our system is mostly based on this source. We are using two approaches for people localization—people appearance modeling and face recognition. People appearance modeling is based on color features. An object can have several color models—one or more for each location or even for the time of the day. Object models can be defined (through training) prior to the surveillance task or accumulated incrementally during the task. Appearance modeling works for all cameras, whereas face recognition is efficient only for some cameras where size and orientation of faces are appropriate. We use a dedicated PTZ camera that watches the main entrance to the floor for face recognition. The face recognition system uses the OpenCV algorithm [15] and tries to recognize people from a restricted list.

21.5.1.2 Infrared Badge ID System

The second source of evidence is the infrared (IR) badge system. The system collects data from 91 readers and merges them into a database that indicates where a particular badge was sensed the last time. This source of information is not very reliable because of the following reasons.

1. The badge has to be in the line of sight of a reader on the ceiling. If a person puts his or her badge into a pocket, it cannot be detected.
2. The orientation of the badge affects the detection. A person may be standing under an IR-reader but his badge could trigger another reader nearby depending on the orientation of the badge.
3. A person can leave his or her badge in the office or give it to another person.
4. Detection records are written to the database with a delay creating a discrepancy among different sources of evidence for fast moving objects.

Before processing IR badge sensor signals, the system must determine and maintain the list of active sensors, which are sensors that both transmit signals and move in space.

21.5.1.3 Finger Print Reader

The third source of evidence is the fingerprint reader. This is a very reliable source, but located only at the main entrance, has a restricted number of registered users, and a person only uses it 1–2 times per day for check-in. We mostly use it for acquisition or updating of person appearance models as a person checks-in when entering the office.

21.5.1.4 Human Intervention

The fourth source of evidence is human intervention. People who participate in a surveillance task can interactively influence the system. They can mark an object in a camera view and associate it with a particular person, which causes the system to set the probability of the person being at this location to 1 and recalculate the previous decisions by tracking the person back in time. This is a very reliable, but costly information source. We use it mostly for initializing and updating person appearance models.

21.5.2 Identification and Tracking of Objects

The current state of the world is specified by a probability distribution of objects being at particular locations at each time tick. Let us assume that $P(H_i \mid L_j), i = 1, N, j = 1, K$ are probabilities to find the object H_i at location L_j. The initial (prior) distribution can be learned from data or assumed to be uniform. Each object has a set of models that are location and sensor specific.

21.5.2.1 One Sensor, One Location

Suppose we have only one sensor (camera or IR badge reader). It senses one location, captures events, and saves them in the event database. For the camera, an event is a time-ordered sequence of frames with motion. For an IR reader, an event is a sequence of sets of people IDs detected at this location. Taking into account that the IR badge system gives the list of people IDs directly, we concentrate first on processing data

from a camera, and then consider how to merge the decisions from both sensors when they are available.

The task is to identify people whom the camera sees. We assume that we have models for each of N people. We also assume that we have prior probabilities $P(H_i)\ i = 1, N$ for the person i being in front of the camera. The prior probabilities can be estimated from data or assumed to be equal if no data are available.

The processing agent performs the following algorithm for each event.

Step 1. For the current frame extract regions that correspond to people (objects). The result is a set of regions (blobs) $\boldsymbol{R} = \{R_j\} j = 1, M$.

Step 2. For each region R_j do the following.

2.1 Estimate likelihoods $P(R_j \mid H_i)\ i = 1, N$ of the region to belong to the model of person H_i.

2.2 If all likelihoods are below a threshold Th, then the blob represents an unknown person. In case when the system tracks all people, it creates a new ID and a model for this person, otherwise it marks the blob as "unknown."

2.3 If one or more likelihood is above the threshold, calculate posterior probabilities using Bayes formula,

$$P\left(H_i \mid R_j\right) = \frac{P\left(R_j \mid H_i\right) \cdot P\left(H_i\right)}{P\left(R_j\right)} \tag{21.4}$$

where $P\left(R_j\right) = \sum_{i=1}^{N} P\left(R_j \mid H_i\right) \cdot P\left(H_i\right)$ is the complete probability of the region R_j.

2.4 Assign to the blob a person that maximizes the posterior probability. Exclude this person from the list for the other regions (a person cannot be represented by more than one blob). Pick up another blob, go to Step 2.1.

Step 3. Do steps 1–2 for each frame of the event. There are several ways of how to process the next frame. One of them is to use the same initial prior probability distribution for each frame. In this case we consider each frame independently (a "bag" of frames approach), and the final result does not depend on the sequence of frames. This approach can be more robust when occlusions occur. If an occlusion happens, the system just loses a blob for the current frame. But this gap can be restored at the postprocessing step using median filtering on the probability sequence for the occluded person. Another approach is using current probabilities as the prior probabilities. In this case the frames are processing consequently in forward or backward direction. This approach requires some heuristics in case of occlusions, such as keeping the probability for disappeared blob unchanged.

Step 4. Do postprocessing. It includes applying smoothing procedures, such as median filtering of the probability sequences of detected people and summarizing result for the whole event.

When merging evidence from camera and IR badge sensors, the system takes into account the peculiarities of IR badge sensor mentioned above. First, it shifts data to compensate for 3-s delay of the IR badge signal. Second, it uses signals only from active sensors. Third, in spite of binary evidence the system uses a likelihood

function that gives the probability of 0.95 for the people on the evidence list and low probability for all other people (see below for more elaborated likelihood function for multisensor and multilocation case), which is used in Equation (21.4). If no evidence came in the next tick, the likelihood function degrades exponentially.

21.5.2.2 Multiple Sensors and Multiple Locations

In this case we have to deal with new challenges such as synchronization of multiple sensors in time and space.

On one hand, some sensors such as video cameras can have large fields of view that can be divided into nonoverlapping locations. On the other hand, the sensing zones of different sensors can intersect. These intersections can be considered as natural locations. Sometimes the borders of locations are fuzzy.

The concept of location allows making more precise localization, and having person models for each location to improve person identification. On the other hand, we have to create person models for each person and for each location that often is not possible because the person cannot visit all locations during the day. That is why we assume that a person may have models only for some locations. If a person does not have a model for the location under consideration, then the model for the closest location is used.

Each person also has a transition matrix $T(H_k) = \{t_{ij}(H_k)\} k = 1, N, i, j = 1, K$, that specifies the probability of person transition from i-th to j-th location.

The major concern for a multisensor environment is accuracy of data synchronization. Different sensors may have different sampling rates and can acquire signals in nonregular intervals. In general, surveillance cameras are not synchronized. However, computers' clocks can be synchronized and time stamps can be assigned to frames. This means that we cannot synchronize frames, but we can select frames that belong to time interval of some duration (time tick). The time tick should be big enough to contain at least one frame from each camera, but be small enough to allow people moving only inside the current location or to the one of adjacent locations. In our experiments time tick is equal to 1 s.

Another issue is that sensors of the same or different type can have overlapping sensing zones. It poses some restrictions on locations form and size. For example, Figure 21.4 shows two cameras with overlapping field of views. The area has six locations. Locations around the overlap (L_2, L_3, L_4) have more sophisticated geometry. The graph on the right represents transitions among locations. (Here locations HL_1–HL_3 correspond to hidden areas.)

The process of identification and tracking of objects consists of the following steps.

Step 1. Data Collection and Feature Extraction: Collect data from all sensors related to the same time tick. Select data that contain information about a new "event" and extract features.

Step 2. Object Unification From Multiple Sensors: Each sensor detects signals of one or more objects in its sensory field. The signals that come from the same object are merged on the basis of their location and sensory attributes. This gives us a

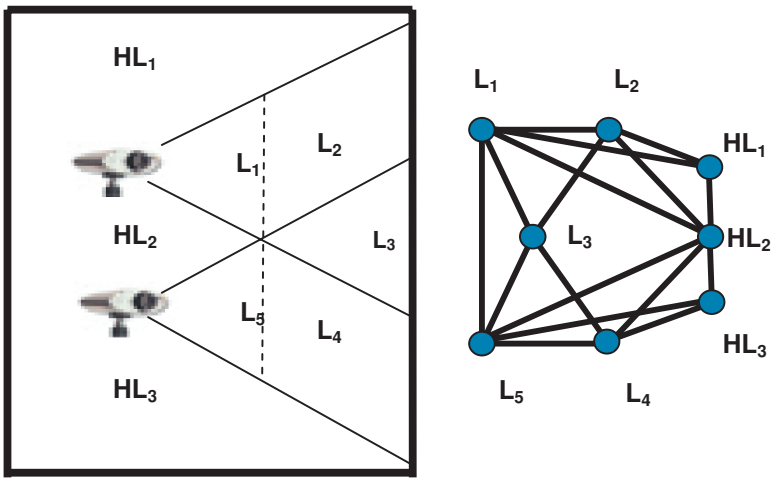

Fig. 21.4. Observed and hidden locations for two cameras.

unified model of how different sensors "see" the same entity. For video cameras, the blobs are first mapped into locations based on their coordinates and calibration data from the cameras. Then the blobs from different cameras that belong to the same location are assigned to the same entity based on their color features. For IR badge data, which consist of binary indicators of a badge being detected at a particular location, the system first spreads the probability to the adjacent IR locations taking into account the space geometry, and then maps IR locations into camera-based locations and associates evidence with entities. The result is a set of entities $\boldsymbol{O} = \{O_r\}$ and a matrix $\boldsymbol{W} = \{w_{kr}\}k = 1, K, r = 1, M_0$, where M_0 is the number of entities. Each w_{kr} is the membership value of r-th entity to belong to the k-th location.

Step 3. Motion Estimation: The locations are selected in a way that an object can either stay in the same location or move to an adjacent location during any single time tick. The specific transition probabilities among locations for a known object or generalized transition probabilities for the other objects are estimated from historical data or provided as prior knowledge by the people involved in the task. These probabilities are taken into account for re-estimating prior probabilities using Equation (21.5).

$$\tilde{P}(H_i \mid L_j) = \frac{\left[\sum_{k=1}^{L} P(H_i \mid L_k) \cdot t_{kj}(H_i)\right] \cdot P(H_i \mid L_j)}{\sum_{l=1}^{L}\left[\sum_{k=1}^{L} P(H_i \mid L_k) \cdot t_{kj}(H_i)\right] \cdot P(H_i \mid L_l)} \tag{21.5}$$

This is a kind of motion prediction in case when we do not know anything about the person's movement except that he or she was previously in a particular location. Adding more information to a person's state, such as direction of movement,

velocity, acceleration, etc., makes possible applying more advanced tracking techniques, such as Kalman filtering [16], particle filtering [17], or Bayesian filtering [18].

Step 4. Posterior Probability Estimation: Using the features that belong to the same entity and the person models, the conditional probability that the entity represents a person at a given location is estimated for all entities, objects, and locations. The result is a sequence of probabilities $S_r = \{P(R_j, L_k, C_q \mid Hi,)\}$ associated with the entity $O_r.r = 1, M_0$. Here R_j, $j = 1, M_r$ are the feature data extracted from representations of entity O_r, and $C_q, q = 1, Q$, are sensors. For video cameras, the probabilities that a blob represents an object (person) for given cameras and locations are calculated using blob's features and persons' models (see Equation (21.3)). For IR badge data the probabilities distributed to adjacent locations are used as the conditional probabilities. The output of face recognition system is also used as conditional probabilities. The fingerprint and human intervention evidence sets up the prior probabilities directly. The difference between video sensors and the other sensors is that the data from video cameras are used for every tick, but the data from the other sensors are used only when they are available.

For each entity the estimates of likelihood that the entity represents a particular person at a given location are calculated. If all estimates are less than a threshold, then the entity is marked as "unknown," and a new ID and a new model are generated. Otherwise, the conditional probabilities of signals that are views of the same entity from different sensors are used for estimating posterior probabilities of a person being represented by the entity at the location using Bayes rule (21.6) and the person's ID that maximizes the conditional probability is assigned to the entity. Then the model of the just assigned person is excluded from the model list for processing the other entities.

$$P\left(H_i \mid O_r, L_k\right) = \frac{\tilde{P}\left(H_i \mid L_k\right) \cdot w_{kr} \cdot \prod\limits_{P(R_j, L_k, C_q) \in S_r} P\left(R_j, L_k, C_q \mid H_i\right)}{P\left(O_r\right)} \tag{21.6}$$

where $P\left(O_r\right) = \sum\limits_{i=1}^{N} \tilde{P}\left(H_i \mid L_k\right) \cdot w_{kr} \cdot \prod\limits_{P(R_j, L_k, C_q) \in S_r} P\left(R_j, L_k, C_q \mid H_i\right)$.

Then the probabilities for the entity are normalized over locations using (21.7).

$$P(H_i \mid L_k) = \frac{P(H_i \mid O_r, L_k)}{\sum\limits_{k=1}^{L} P(H_i \mid O_r, L_k)} \tag{21.7}$$

Step 5. Reestimation: Steps 1–4 are repeated for each time tick.

Step 6. Postprocessing: This step includes some smoothing procedures for whole events and truth maintenance procedures, which use problem domain knowledge to maintain probabilities when no data are available. In case when an object is temporarily invisible, the truth maintenance procedures mark it as "idle" and keep its probability high to be in "hidden" locations that are near the location where the object has been identified last time. For example, Figure 21.4 shows two cameras that watch a room. There are five observed locations (L_1–L_5) and three hidden

locations (HL_1–HL_3). The graph on the right shows transitions among locations. If an object has been seen last time at location L_2 then there is high probability for the object to be in the location HL_1, but if no additional evidence is available the probabilities to be in locations HL_2 and HL_3 are also growing over time until all three became equal.

The classical Bayesian approach assumes that (1) there are a constant number of mutually exclusive hypotheses, and (2) the hypotheses cover the whole decision space. In our case the situation is more dynamic—people may enter and leave the floor, and people may be "invisible" to the sensors, for example, a person has his IR badge covered and is standing in a "dead zone," which is not observed by any camera. We extended the framework to cover these problems. The system uses two cameras that watch the elevator area and detects people who are entering or leaving the floor. If a person leaves the floor, his or her model are marked as "inactive." If a person enters the floor, a new object and its appearance model are created and are marked as "new." The system tracks a new object and creates models for it for other locations when it is possible (high probability of identifying the person, no occlusion, etc.). The process continues until enough data is collected.

21.6 Experimental Results

For evaluation we used 15 cameras and 44 IR badge readers that are located in the northern half of the floor. In the first experiment we evaluated the system's performance in the closed set case. It means that the system had models for all 15 people, who participated in the experiment. The second experiment was designed for evaluating the system's performance for the open set problem. Besides 15 "known" people it included 10 "unknown" people, that is, people whose models were not available at the beginning and created during the process. Each experiment lasted for 4 h from 10 AM till 2 PM. In both experiments two evaluations have been done. The first evaluation estimated the accuracy of people localization for each camera separately, and then calculated the average for each person. The second evaluation merged the results from all cameras and IR badge readers. It is worth to mention that only 7 of 15 "known" people (marked by stars in the Table 21.2) and none of "unknown" had active IR badges. The results have been compared to the ground truth data created manually for each tick.

Table 21.2 presents results for both experiments. From the table and Figure 21.5 we can see that in case of closed set problem the average recognition accuracy is about 11% higher than for the open set problem for both single camera and integrated evaluations. For the closed set problem the accuracy of localization for individual person is in range from 44% to 99%. The low accuracy for some people can be explained by the following:

- Poor blob extraction for people who are sitting still for a long time.
- Poor blob extraction when a person is (partially) occluded (e.g., a person was sitting in his office partially visible through the door).

Table 21.2. Accuracy of people localization for open and closed set problems.

	Closed set accuracy			Open set accuracy		
PersonID	Single Camera	Cameras and IR badge	Diffe-rence	Single Camera	Cameras and IR badge	Difference
1000	82.14%	87.56%	5.42%	71.39%	77.31%	5.92%
1002	99.27%	99.51%	0.24%	94.54%	95.81%	1.27%
*1003	86.88%	91.07%	4.19%	81.91%	86.10%	4.19%
1005	74.32%	81.69%	7.37%	62.73%	70.59%	7.86%
*1006	45.98%	69.68%	23.70%	43.78%	58.58%	14.80%
*1015	44.08%	41.12%	−2.96%	26.32%	26.58%	0.26%
1020	67.03%	76.71%	9.68%	60.75%	69.24%	8.49%
*1023	64.43%	60.77%	−3.66%	59.82%	58.49%	−1.33%
1024	41.26%	50.72%	9.46%	25.08%	32.78%	7.70%
*1025	69.26%	78.29%	9.03%	57.88%	66.81%	8.93%
1026	71.06%	73.66%	2.60%	41.34%	46.19%	4.85%
*1027	62.04%	73.41%	11.37%	58.13%	67.08%	8.95%
*1029	51.21%	57.88%	6.67%	51.50%	57.21%	5.71%
1064	77.81%	83.42%	5.61%	44.69%	51.69%	7.00%
1072	66.07%	74.49%	8.42%	52.53%	59.67%	7.14%
Average	66.86%	73.33%	6.47%	55.49%	61.61%	6.12%

- Poor blob separation in the hallways.
- Several people have similar models.

Merging evidence from several cameras and IR sensors improves the performance for both cases for about 6%. For the closed set problem the largest improvement (23.7%) was mostly due to IR badge data. In this case, a person ID1006 stayed in the same location for a long time with his badge active. But sometimes merging IR badge

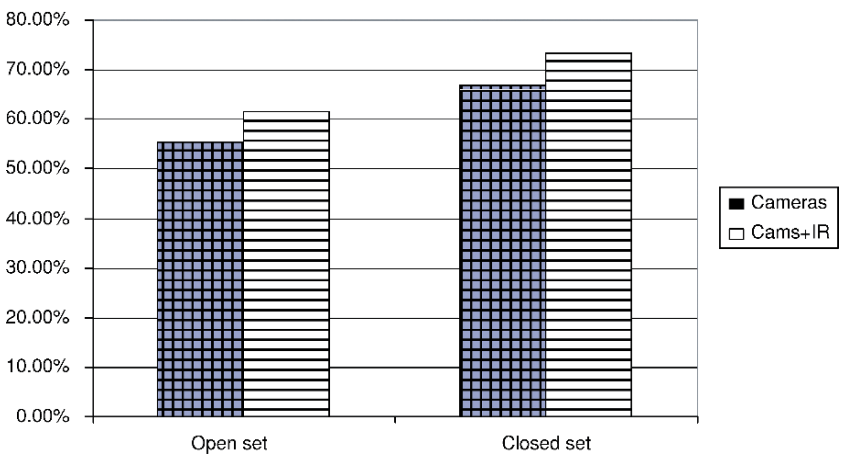

Fig. 21.5. Average accuracy of people localization.

data causes a decrease of localization accuracy. It happens for transient events in the hallways because of poor alignment of visual and IR data (see results for ID1015 and ID1023). The increase from merging visual evidence from several cameras can reach up to 9%.

For the open set problem the localization accuracy for individual person lies in the range from 25% to 94%. Low accuracy for some people can be mostly attributed (besides the above-mentioned reasons) to confusion with people who have similar models. Merging additional evidence can improve performance up to 15%. Merging only visual evidence from several cameras improves performance by 7–8%.

21.7 Summary

In this chapter we described a Bayesian framework that enables us to robustly reason from data collected from a network of various kinds of sensors. In most practical situations, sensors are producing streams of redundant, but noisy data. We proved experimentally that the probabilistic framework presented here gives us the ability to reason from this data by also incorporating the local semantics of the sensors as well as any domain knowledge that can be provided by people involved in these tasks. We believe that this framework is applicable in the larger context of creating robust and scalable systems that can reason and make inferences from different kinds of sensors that are present in the world today.

As to future work, we see that the system could be improved on many levels. On the low level it needs more robust background modeling, blob extraction and blob separation techniques, search for better features, and reliable dynamic modeling of people and other objects' appearance. On the middle level it needs using more advanced tracking approaches such as nonlinear filtering [17, 18] and sensorial data fusion approaches [19]. On the high level the system needs more efficient decision merging approach, which can use domain-specific knowledge and can produce a consistent "big picture" of events in the area under surveillance. We also plan to spend time for developing more attractive visualization techniques and a useable user interface.

References

1. Fuentes, LM, Velastin, SA. People tracking in surveillance applications. In: *Proc. 2nd IEEE International Workshop on PETS*, Kauai, Hawaii, USA, December 2001.
2. Siebel, NT, Maybank, S. Fusion of Multiple Tracking Algorithms for Robust People Tracking. Proc. 7th European Conference on Computer Vision (ECCV 2002), Copenhagen, Denmark, May 2002; IV:373–387.
3. Krumm, J, Harris, S Meyers, B Brumitt, B Hale, M Shafer, S. Multi-camera multi-person tracking for easy living. In: *Proc. 3rd IEEE International Workshop on Visual Surveillance*, July 1, 2000, Dublin, Ireland.

4. Mittal, A, Davis, LS. M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 2003; 51(3):189–203.
5. Kettnaker, V, Zabih, R. Bayesian multi-camera surveillance. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, June 23–25,1999;2253–2259.
6. Cai, Q, Aggarwal, JK. Tracking human Motion using multiple cameras. In: *Proc. International Conference on Pattern Recognition*, Vienna, Austria, August 1996:68–72.
7. Cai, Q, Aggarwal, JK. Tracking human motion in structured environments using a distributed-camera system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1999) 2(11):1241–1247.
8. Khan S, Javed O, Rasheed Z, Shah M. Human tracking in multiple cameras. In: *Proc. 8th IEEE International Conference on Computer Vision*, Vancouver, Canada, July 9–12, 2001;1:331–336.
9. Javed O, Rasheed Z, Atalas O, Shah, M. KnightM: A real time surveillance system for multiple overlapping and non-overlapping cameras. In: *The fourth IEEE International Conference on Multimedia and Expo (ICME 2003)*, Baltimore, MD, July 6–9, 2003.
10. Huang KS, Trivedi MM. Distributed video arrays for tracking, human identification, and activity analysis. In: *The fourth IEEE International Conference on Multimedia and Expo (ICME 2003)*, Baltimore, MD, July 6–9, 2003;2:9–12.
11. Cheung S-CS, Kamath Ch. Robust techniques for background subtraction in urban traffic video. In: *Proc. of SPIE, Visual Communications and Image Processing 2004*, S. Panchanathan, B. Vasudev (Eds), January 2004;5308:881–892
12. Toyama K, Krumm J, Brumitt B, Mayers B. Wallflower: Principles and practice of background maintenance. In: *Intl Conference on Computer Vision (ICCV)*, 1999;255–261.
13. Stauffer C, Grimson WEL. Adaptive background mixture models for real-time tracking. In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999; 246–252.
14. Power PW, Schoonees JA. Understanding background mixture modeling for foreground segmentation. In: *Proc. Image and Vision Computing*, New Zealand, 2002;267–271.
15. Nefian AV, Hayes III. M.H. Maximum likelihood training of the embedded HMM for face detection and recognition. *IEEE International Conference on Image Processing*, September 2000;1:33–36.
16. Grewal MS, Andrews AP. *Kalman Filtering. Theory and Practice Using Matlab*. John Wiley & Sons, 2001.
17. Ristic B, Arulampalam S, Gordon N. *Beyond the Kalman Filter. Particle Filters for Tracking Applications*. Artech House: Boston, London, 2004.
18. Stone LD, Barlow CA, Corwin TL. *Bayesian Multiple Target Tracking*. Artech House: Boston, London, 1999.
19. Hall DL, McMullen SAH. *Mathematical Techniques in Multisensor Data Fusion*. Artech House: Boston, London, 2004.